Finding all minimal CURB sets

Max Klimm and Jörgen W. Weibull

SSE/EFI Working Paper Series in Economics and Finance, No 722

December 9, 2009

Abstract

Sets closed under rational behavior were introduced by Basu and Weibull (1991) as subsets of the strategy space that contain all best replies to all strategy profiles in the set. We here consider a more restrictive notion of closure under rational behavior: a subset of the strategy space is *strongly* closed under rational behavior, or sCURB, if it contains all best replies to all probabilistic beliefs over the set. We present an algorithm that computes all minimal sCURB sets in any given finite game. Runtime measurements on two-player games (where the concepts of CURB and sCURB coincide) show that the algorithm is considerably faster than the earlier developed algorithm, that of Benisch et al. (2006).

Keywords: Curb Sets; Algorithm; Rational Behavior; Rationalizability; Minimality.

JEL codes: C02, C62, C63, C72.

1 Introduction

Since the pioneering works of John von Neumann, Oskar Morgenstern and John Nash, non-cooperative games have become the standard framework for analyses of a very wide range of strategic interactions in economics and other social and behavioral sciences. Despite the enormous range of applications of Nash equilibrium as a solution concept, its stability has been queried in the works of Selten (1975) and Myerson (1978) and others, leading to a range of refinements such as perfect and proper equilibria. Any strict Nash equilibrium (one in which each player's strategy is her unique best reply) satisfies these refinements. However, many games have no strict equilibria. A set-valued version of strict equilibrium was proposed by Basu and Weibull (1991). They call a set of strategy profiles *closed under rational behavior*, or CURB for short, if it contains all best replies to itself. Minimal such sets exist in a large class of games, including all finite games.

Also other set-valued solution concepts have been introduced in the game-theory literature, such as *persistent retracts* by Kalai and Samet (1984), *strategically stable sets* by Kohlberg and Mertens (1986) and *preparation sets* by Voorneveld (2004). As shown in Voorneveld (2005), the unique product-set valued solution concept that satisfies *nonemptiness*, *one-person rationality*, *consistency*, *nonnestedness*, and *satisfaction* is the minimal CURB concept. Another reason to study minimal CURB sets is that they are attractors to several processes of social learning, see the works of Young (1998) and Hurkens (1995).

We here present an algorithm that identifies all minimal CURB sets in any given finite two-player game. It also finds all minimal strong CURB sets in any finite game at all. The distinction between CURB and strong (or correlated) CURB is immaterial in two-player games but matters for games with more players. While CURB is based on the Nash paradigm of uncorrelated individual strategies, strong CURB allows a player to believe that other players' actions may be correlated (statistically dependent).

In the history of algorithmic work in game theory, that of Lemke and J. T. Howson is a land-mark. They proposed an algorithm that finds all Nash equilibria in any given finite two-player game. However, it was shown by Savani and von Stengel (2004) that there are such games for which the runtime of this algorithm is exponential in the size of the game (defined as the total number of pure strategies). Moreover, the problem of finding one Nash equilibrium in such games is known to be PPAD-complete, suggesting that no polynomial algorithm exists. This holds even when the payoffs received by the players are restricted to be binary, see Chen and Deng (2006) and Abbott et al. (2005).

Concerning CURB sets, the first algorithm was provided by Pruzhansky (2003), who proposed an algorithm that identifies CURB sets in finite games in extensive form of perfect information. Such games possess a unique minimal CURB set, and this can be computed in a relatively straightforward manner. Several algorithms to compute CURB sets in finite two-player games in normal (or strategic) form was developed by Benisch et al. (2006). The present algorithm builds in part on their work and we conclude by comparing the performance of their and our algorithms.

The first step in our analysis concerns the geometric structure of so-called *stability sets*, introduced by Harsanyi and Selten (1988). These are pre-images of the pure best-reply correspondence. We show that, in finite two-player games, all stability sets are polytopes.

For finite games with three or more players, the stability sets are not always polytopes. In fact, they are not even convex in general. This is due to the assumed statistical independence between different players' randomization. As a consequence, stability sets cannot be defined by linear inequalities, in general. For games with more than two players, we overcome this inconvenience by allowing each player to believe that others' actions are correlated (or dependent), and we modify the definition of stability sets accordingly, obtaining what we call *strong stability sets*. The strong stability set, for a given pure strategy of a player in a finite game, is the set of probability distributions over the others' pure-strategy combinations, under which the pure strategy in question is optimal. So defined, strong stability sets are polytopes, in all finite games. We use the notion of strong stability sets to define *strong CURB sets*, or sCURB sets for short, requiring that they contain all best replies to themselves, without the restriction that each player believes that the others' actions are statistically independent. The same approach is taken in Asheim et al. (2009), who provide epistemic characterizations of sCURB sets in finite games. Sets of this nature were first introduced by Harsanyi and Selten (1988), in their analysis of the agent normal form of finite extensive-form games. The called such sets *primitive formations*.

Our main results concern an algorithm that we propose. This algorithm computes all minimal sCURB sets in any given finite game. We show that this is possible to do by way of solving certain linear feasibility problems (LFPs). The algorithm starts out from a certain set of candidate sCURB configurations and checks the sCURB property iteratively. Each group of LFPs either approves the sCURB property for a particular candidate or, if not approved, increases the size of the candidate configuration by successively adding pure strategies. The algorithm terminates in finite time and identifies all sCURB sets.

As the concepts of CURB and sCURB coincide for two-player games, we are able to compare the performance of our algorithm with the one proposed by Benisch et al. (2006). We also compare our computer simulation results concerning the size distribtution of minimal sCURB sets with a theoretical result for Nash equilibria, due to Dresher (1970). In particular, an adaptation of that result to the present setting implies that the probability that a finite two-player game with randomly drawn payoffs will have a singleton sCURB set (a strict Nash equilibrium) converges to $1 - 1/e$ as the number of pure strategies of both players tends to plus infinity.

Part of this work is based on a scientific internship report at École Polytechnique, see Klimm (2008).

2 Preliminaries

Consider finite normal-form games $G = (N, S, u)$, where $N = \{1, ..., n\}$ is the non-empty and finite set of players, $S_i = \{1, \ldots, m_i\}$ the non-empty and finite set of pure strategies available to player $i \in N$, $S = \times_{i \in N} S_i$ the set of pure-strategy profiles $s = (s_1, ..., s_n)$, and $u : S \to \mathbb{R}^n$ is the combined payoff function that assigns a payoff $u_i(s) \in \mathbb{R}$ to each pure-strategy profile $s \in S$ and player $i \in N$. The total number of pure-strategy profiles in the game is $m = m_1 \cdot ... \cdot m_n$.

To allow for randomizations, we identify each pure strategy $h \in S_i$ with the h:th unit vector in R^{m_i}, that is, with the mixed strategy that assigns unit probability to that pure strategy, and define the *linear (or mixed-strategy) extension* of the game G as the game $\tilde{G} = (N, \square(S), \tilde{u})$, where the product set $\square(S) = \times_{i \in N} \Delta(S_i)$ is the polyhedron of mixed-strategy profiles $x = (x_1, ..., x_n)$, with $\Delta(S_i)$ denoting the set of mixed strategies available to player i; the unit simplex in $\mathbb{R}^{m_i}$ spanned by player i's pure strategies (viewed as unit vectors). The combined mixed-strategy payoff function $\tilde{u} : \square(S) \to \mathbb{R}^n$ is defined by $\tilde{u}_i(x) = \sum_{s \in S} \left(\prod_{i \in N} x_i(s_i)\right) u_i(s)\ \forall i \in N$, where $x_i(s_i)$ is the probability that player i uses her pure strategy s_i. We note that $\tilde{u}(x)$ is multi-linear; for each $j \in N$ it is linear in the mixed strategy $x_j \in \mathbb{R}^{m_j}$ (the Euclidean space containing the simplex $\Delta(S_j)$).

For any player i and mixed-strategy profile x, let $x_{-i} \in \square(S_{-i}) = \times_{j \neq i} \Delta(S_j)$ denote the strategy profile of all *other* players, and for any $s_i \in S_i$ and $x_{-i} \in \square(S_{-i})$ let $x' = (s_i, x_{-i})$ denote the mixed-strategy profile in which player i assigns probability one to her pure strategy s_i and the others play according to x. We denote by $\beta_i(x_{-i})$ the set of *pure best replies* of player i to x_{-i}, that is:

$$\beta_i(x_{-i}) = \{s_i \in S_i : \tilde{u}_i(s_i, x_{-i}) \geq \tilde{u}_i(s_i', x_{-i})\, \forall s_i' \in S_i\}.$$

This defines player i's (non-empty valued) best reply correspondence $\beta_i : \square(S_{-i}) \rightrightarrows S_i$. For any subset $X_{-i} \subset S_{-i}$, let $\beta_i(X_{-i}) \subset S_i$ be the direct image of X_{-i} under β_i. (Formally: $\beta_i(X_{-i}) = \cup_{x_{-i} \in X_{-i}} \beta_i(x_{-i})$.) We view $\beta_i(X_{-i})$ as a subset of $\Delta(S_i)$; the collection of unit vectors that correspond to pure best replies to profiles x_{-i} in X_{-i}.

3 Closure and strong closure under rational behavior

Harsanyi and Selten (1988) introduced the notion of *stability sets*. For any player $i \in N$ and pure strategy $s_i = h \in S_i$, the stability set B_{ih} is the pre-image of s_i under i's pure-strategy best-reply correspondence. In other words, the set B_{ih} consists of all those (mixed) strategy profiles $x_{-i} \in \square(S_{-i})$ for which the pure strategy $s_i = h$ is a best reply for player i. Formally:

$$B_{ih} = \{x_{-i} \in \square(S_{-i}) : h \in \beta_i(x_{-i})\} .$$

Kalai and Samet (1984) developed an equilibrium refinement called persistent equilibrium. They did this by way of studying set-valued properties of a class of strategy subsets called *retracts*. A *retract* is a product set $X = \times_{i\in N} X_i$, where each set X_i is a nonempty, closed and convex subset of mixed strategies for player i, $X_i \subset \Delta(S_i)$. Basu and Weibull (1991) call such a product set X *closed under rational behavior* (CURB) if it contains all its best replies, that is, if $\beta_i(X_{-i}) \subset X_i$ for each player i.[1] They call a CURB set X *minimal* if it does not contain any other CURB set.

As shown by Benisch et al. (2006), for linear extensions of finite games, a minimal CURB set X is always a polyhedron, $X = \square(T) = \times_{i\in N}\Delta(T_i)$, for nonempty subsets $T_i \subset S_i$. To see this, suppose that X is a minimal CURB set. Then there exists a unique, nonempty and maximal subset $T_i \subset S_i$ for each player i such that $\Delta(T_i) \subset X_i$, namely $T_i = \beta_i(X_{-i})$. Each set X_i, being convex, contains the convex hull $X'_i = \Delta(T_i)$ of T_i. Moreover, $X' = \times_{i\in N} X'_i$ is a CURB set: $X'_{-i} \subset X_{-i}$ implies $\beta_i(X'_{-i}) \subset \beta_i(X_{-i})$. Hence, minimality implies $X = X' = \square(T)$. Benisch et al. (2006) also show that minimal CURB sets do not overlap. As noted by Basu and Weibull (1991), minimal CURB sets always exist.

However, although minimal CURB sets are mathematically "well-behaved", the defining property of CURB sets is hard to use directly for identification of such sets in games of moderate or large size, since the definition requires computation of all best replies $\beta_i(x_{-i})$ to a continuum of mixed-strategy profiles, $x_{-i} \in X_{-i}$, for each player i. We base our algorithm on the following immediate characterization of CURB sets:

Proposition 1 *Let $\tilde{G} = (N, \square(S), \tilde{u})$ be the linear extension of a finite game. A configuration $T = \times_{i\in N} T_i, T_i \subset S_i$ is a CURB configuration if and only if*

$$B_{ih} \cap \square(T_{-i}) \neq \varnothing \quad \Rightarrow \quad h \in T_i \tag{1}$$

for all $i \in N$ and $h \in S_i$.

[1]Note that a set $X_i \subset \Delta(S_i)$ contains all mixed best replies to a profile x_{-i} if and only if it contains all pure best replies to x_{-i}, since the set of mixed best replies is the subsimplex spanned by all pure best replies.

Equivalently, a configuration $T = \times_{i\in N} T_i$, $T_i \subset S_i$, is *not* a CURB configuration if and only if there is a player $i \in N$ and a pure strategy $h \in S_i \setminus T_i$ such that $B_{ih} \cap \Box(T_{-i}) \neq \varnothing$. From this, it is possible to verify the CURB property of a configuration T by checking, for each player $i \in N$ and pure strategy $h \in S_i \setminus T_i$, whether $B_{ih} \cap \Box(T_{-i})$ is empty. Hence, the algorithmic challenge to verify the CURB property relies on the structure of the stability sets B_{ih}. It is a useful observation — made already by Benisch et al. for the design of their algorithm — that stability sets in finite two-player games are polytopes, that is, they are bounded and convex sets that can be defined by the means of finitely many linear inequalities.

Proposition 2 *In linear extensions $\tilde{G}$ of finite two-player games for every player $i \in N$ and pure strategy $h \in S_i$ the stability set B_{ih} is a polytope in the space $\Delta(S_{-i})$.*

Proof. Let $\tilde{G} = (N, \Box(S), \tilde{u})$ be the linear extension of a finite two-player game. We show the claimed result for the stability sets of of the row player, say player 1, only. To this end, let $h \in S_1$ be arbitrary. Now consider $B_{1h} = \{x_2 \in \Delta(S_2) : h \in \beta_1(x_2)\}$. We may write $x_2 = \sigma = (\sigma_1, \ldots, \sigma_{m_2})$ where σ_k denotes the probability that player 2 plays strategy her k:th strategy. Moreover let $A \in \mathbb{R}^{m_1 \times m_2}$ be defined as

$$A = \begin{pmatrix} u_i(1,1) & u_i(1,2) & \ldots & u_i(1,m_2) \\ u_i(2,1) & u_i(2,2) & \ldots & u_i(2,m_2) \\ \vdots & \vdots & & \vdots \\ u_i(m_1,1) & u_i(m_1,2) & \ldots & u_i(m_1,m_2) \end{pmatrix}.$$

Introducing the convention that $A_{h,\cdot}$ denotes the h:th row of A, the stability set can be written as

$$\begin{aligned} B_{1h} &= \{\sigma \in \Delta(S_2) : A_{h,\cdot}\sigma \geq A_{k,\cdot}\sigma \quad \forall k \in S_1\} \\ &= \left\{ \sigma \in \mathbb{R}^{m_2} : \begin{pmatrix} \sigma_l & \geq & 0 & \forall l \in S_2 & \text{and} \\ \sum_{l \in S_2} \sigma_k & = & 1 & & \text{and} \\ (A_{h,\cdot} - A_{k,\cdot})\sigma & \geq & 0 & \forall k \in S_1 & \end{pmatrix} \right\}. \end{aligned} \tag{2}$$

The second term of Equation (2) is a system of linear equalities and inequalities on σ and thus defines a polytope in $\Delta(S_j)$. ■

As an immediate consequence for two-player games, we obtain that, for any configuration $T = \times_{i\in N} T_i$ (for $T_i \subset S_i$), any player i and pure strategy $h \in T_i$, the intersection $B_{ih} \cap \Box(T_{-i})$ is a polytope in $\Delta(S_{-i})$. We note, however, that the result of Proposition 2 holds only for two-player games. In the three-player game in the following example, none of the stability sets B_{ih} is even convex.

Example 1 (Non-convexity of stability sets) *Consider the game* $\tilde{G} = (N, \square(S), \tilde{u})$ *with three players* $N = \{1, 2, 3\}$ *and strategy spaces* $S_1 = \{U, D\}, S_2 = \{T, B\}, S_3 = \{L, R\}$ *and with the payoff functions shown in Figure 1. This game can be interpreted as a matching pennies game between players 1 and 2 where player 3 acts as a referee deciding who will win when both pennies match.*

	T	B
U	1,-1,0	-1,1,0
D	-1,1,0	1,-1,0

L

	T	B
U	-1,1,0	1,-1,0
D	1,-1,0	-1,1,0

R

Figure 1: Game with non-convex stability sets

As $u_3(s) = 0$ *for all strategy profiles* $s \in S$*, each of player 3's strategies* $h \in S_3$ *is a best reply to all strategy profiles* $x_{-3} \in \square(S_{-3})$ *and thus* $B_{3h} = \square(S_{-3})$*. We will show now that* $\square(S_{-3})$ *is not convex in* $\Delta(S_{-3})$*. For this, we calculate* $\square(S_{-3}) = \{\gamma U + (1-\gamma)D : \gamma \in [0,1]\} \times \{\delta T + (1-\delta)B : \delta \in [0,1]\}$*. In particular,* $(U,T), (D,B) \in \square(S_{-3})$*. However, it is easy to check that the mixed strategy profile* $(U,T)/2 + (D,B)/2 \notin \square(S_{-3})$*. We derive that* $\square(S_{-3})$ *is not convex. The projection of* $\square(S_{-3})$ *to* $\Delta(S_{-3})$ *is depicted in Figure 2 and shows that* $\square(S_{-3})$ *is not even linear.*

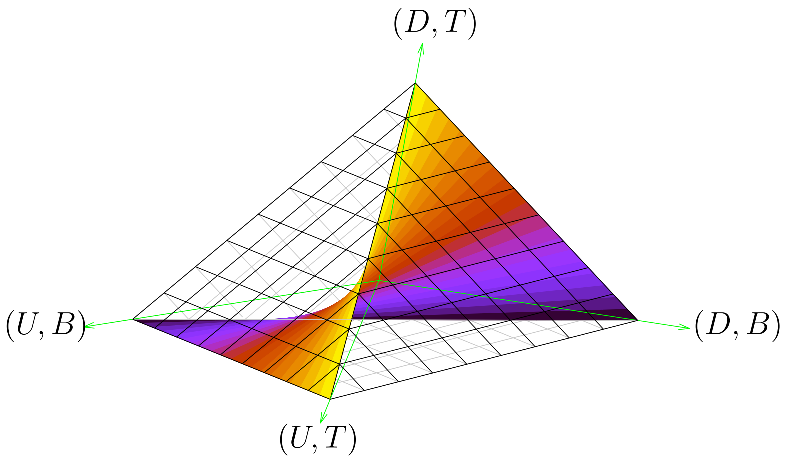


Figure 2: Stability sets of player 3 in the game of Figure 1

Moving to games with more than two players, we note the distinction between, on the one hand, mixed-strategy profiles $x_{-i} \in \square(S_{-i})$ for all other players than i, and, on the other hand, arbitrary probabilistic beliefs $\mu^i \in \Delta(S_{-i})$ that player i may hold over other players' pure strategy choices. This distinction is evidently immaterial for two-player games, but matters for games with more players. For while a strategy profile x_{-i} assumes statistical independence between all other player's actions, μ^i does not. Mathematically, while x_{-i} is the product of probability distributions, one for each pure-strategy set S_j (for each $j \neq i$), a belief $\mu^i \in \Delta(S_{-i})$ is a probability distribution over the set $S_{-i} = \times_{j\neq i} S_j$ of others' pure-strategy profiles:

$$\Delta(S_{-i}) = \left\{ y \in \times_{j\neq i}\mathbb{R}_+^{m_j} : \sum_{j\neq i}\sum_{k\in S_j} y_{jk} = 1 \right\}$$

A point that was already made by Aumann (1974, 1987) is that, in many applications of game theory, the statistical independence of players' actions, assumed in the definition of Nash equilibrium, is not realistic. For instance, players may condition their action on the observation of some external signal, or on a signal that depends on past action profiles, actions taken by other players when playing the same game (as in dynamic social learning processes). Then players' current actions may be correlated. While we do not want to here model such signals explicitly, we do want to allow players to form arbitrary probabilistic beliefs over other players' actions. For this purpose, we introduce the notion of strong stability sets. For each player i and each of her pure strategies $h \in S_i$, this is the subset of probabilistic beliefs $\mu_i \in \Delta(S_{-i})$ under which strategy h is optimal.

Definition 1 (Strong stability set) *Let $\tilde{G}$ be the linear extension of a finite game. For a player $i \in N$ and her pure strategy $h \in S_i$ we call $\hat{B}_{ih} = \{\mu^i \in \Delta(S_{-i}) : h \in \beta_i(\mu^i)\}$ the* strong stability set *of player i and pure strategy h.*

Clearly, for two-player games, this definition coincides with that for stability sets. The definition of strong stability sets essentially says that player i views all other players as a single player with pure-strategy set S_{-i}. Thus, the strong stability sets of a player i in an arbitrary finite game coincide with the stability sets in a two-player game $\hat{G}_i = \left(\hat{N}, \hat{S}, \hat{u}\right)$ where $\hat{N} = \{i, 0\}$, $\hat{S} = \hat{S}_i \times \hat{S}_0$ for $\hat{S}_i = S_i$ and $\hat{S}_0 = \times_{j\neq i} S_j$, with $\hat{u}_i = u_i$ and an arbitrary payoff function u_0. The strong stability sets of player i in the original game G is identical with the stability set of player i in the two-player game $\hat{G}_i$. It follows from Proposition 2 that strong stability sets, $\hat{B}_{ih} \subset \Delta(S_{-i})$, are polytopes. We define "strong closure under rational behavior," or *sCURB*, by means of these strong stability sets.[2]

[2]This route is also taken in Asheim et al. (2009).

	T	B
U	2,2,2	2,2,0
D	2,2,0	2,2,0

L

	T	B
U	1,1,1	-1,-1,-1
D	-1,-1,-1	1,1,1

C

	T	B
U	2,2,0	2,2,0
D	2,2,0	2,2,2

R

Figure 3: Game where sCURB sets and CURB sets do not coincide

Definition 2 (Strong closure under rational behavior) *Let $\tilde{G} = (N, \Box(S), \tilde{u})$ be the linear extension of a finite game. A configuration $T = \times_{i \in N} T_i, T_i \subset S_i$ is called* strongly closed under rational behavior*, mnemonic sCURB, if for all players $i \in N$ and pure strategies $h \in S_i \setminus T_i$ it holds that $\hat{B}_{ih} \cap \Box(T_{-i}) = \varnothing$.*

Clearly, minimal sCURB sets always exist in finite games. Moreover, a sCURB set is necessarily also a CURB set, and thus each minimal sCURB set contains at least one minimal CURB set. The following example shows that not all CURB sets are sCURB.

Example 2 (Minimal CURB sets and minimal sCURB sets) *Consider the linear extension $\tilde{G} = (N, \Box(S), \tilde{u})$ in Figure 2 where $S_1 = \{U, D\}, S_2 = \{T, B\}, S_3 = \{L, C, R\}$. For the dependent belief of player 3 that player 1 and 2 play $\frac{1}{2}\{U, T\} + \frac{1}{2}\{D, B\}$ the center strategy leads to a payoff equal to 2 as the other two strategies do. Thus the minimal sCURB set comprises the center strategy of player 3. In fact one can compute that the minimal sCURB set equals $\{U, D\} \times \{T, B\} \times \{L, C, R\}$ In contrast, there is no independent strategy profile $x_{-3} \in \Box(S_{-3})$ such that $C \in \beta_3(x_{-3})$ and hence the minimal CURB set equals $\{U, D\} \times \{T, B\} \times \{L, R\}$.*

Remark 1 *Note the parallel with the distinction in game theory between rationalizability and correlated rationalizability. Pearce (1984) showed that for pure strategies in finite two-player games, being a best reply to some probabilistic belief about the other player's action is equivalent to not being strictly dominated, while for games with more than two players, this equivalence holds only if players' are allowed to believe in correlation between the others' actions.*

4 The algorithm

In this section we present our algorithm, that identifies all minimal sCURB sets in any given finite game. This is obtained in two steps. In the first step we construct a suitable family

$\mathcal{T}$ of configurations $T = \times_{i \in N} T_i$ (for $T_i \subset S_i$) that is large enough to contain all sCURB configurations. In force of Proposition 1, it is necessary and sufficient for a configuration to be sCURB that the polytope $\hat{B}_{ih} \cap \Delta(T_{-i})$ is empty for all players i and all pure strategies $h \in S_i \setminus T_i$. Checking the emptiness of a polytope is a linear problem that can be solved in polynomial time of low order, see Ye (2006).

Given the family T of configurations, in the second step of the algorithm each candidate set T from the family $\mathcal{T}$, is picked out and certain linear feasibility problems are solved in order to determine whether T is sCURB or not. If T is not sCURB, then T is enlarged until it becomes sCURB (recall that S is sCURB). For this procedure to work, the family $\mathcal{T}$ of initial configurations has to be complete in the following sense ($\mathcal{P}(S)$ denotes the power set of the set S):

Definition 3 (Sub-completeness with respect to sCURB sets) *Let $\tilde{G} = (N, \square(S), \tilde{u})$ be the linear extension of a finite game. A family of non-empty subsets $\mathcal{T} \subset \mathcal{P}(S)$ is called* sub-complete with respect to sCURB sets *if for every minimal non-empty sCURB set C there is a set $T \in \mathcal{T}$ such that $T \subset C$.*

The family $\mathcal{P}(S)$ is itself sub-complete with respect to sCURB sets, since every non-empty sCURB set contains at least one pure strategy combination $s \in S$. We will take the family $\mathcal{T}$ to be those configurations $T = \times_{i \in N} T_i$ (for $T_i \subset S_i$) that contain all pure best replies to all beliefs that place unit probability on some *pure-strategy* profile in T. We will call these configurations *weak CURB*, or wCURB. Formally:

Definition 4 (Weak CURB configuration) *Let $\tilde{G}$ be the linear extension of a finite game and let $T = \times_{i \in N} T_i, T_i \subset S_i$ be a configuration. T is called a* weak CURB configuration *if for all players $i \in N$ and $h \in S_i \setminus T_i$ it holds that $B_{ih} \cap T_{-i} = \varnothing$.*

Clearly all sCURB configurations are CURB configurarions, and all these are wCURB configurations. It is not difficult to show that the set of minimal wCURB configurations is sub-complete with respect to sCURB:

Proposition 3 *Let $\tilde{G} = (N, \square(S), \tilde{u})$ be the linear extension of a finite game and let $\mathcal{T}$ be the set of minimal wCURB configurations in $\tilde{G}$. Then $\mathcal{T}$ is sub-complete with respect to sCURB sets.*

Proof. It is to show that every minimal sCURB set C contains a minimal wCURB configuration B. C can be written as $C = \square(T), T = \times_{i \in N} T_i, T_i \subset S_i$. As C is closed under best replies it is in particular closed under pure best replies and we may derive that T a wCURB configuration. Now consider the class $\mathcal{T}^C$ of wCURB configurations within C. They can be (partially) ordered by set-wise inclusion $\subset$ establishing that there is at least one set-wise minimal element. This element is a minimal wCURB configuration. ■

The first step in our algorithm is to find all minimal wCURB configurations. We do so by using the so-called (pure-strategy) *best-reply graph*, a construction used by Young in his analysis of social learning processes, see Young (1998). For any finite game $G = (N, S, u)$, this is a graph $\mathcal{G}(G) = (V, A)$ with all pure-strategy profiles as its vertices V. We call two such strategy profiles $s, t \in S = V$ *neighbors* if there is a player $i \in N$ such that s and t differ in the strategy of player i only, that is $s_j = t_j$ for all $j \neq i$. For two such neighbor strategy profiles s and t we insert a directed arc (s, t), from s to t into A, if and only if $t_i \in \beta_i(s_{-i})$. The so constructed directed graph $\mathcal{G}(G)$ is called the best-reply graph of the finite game G. We note that, for any finite game G, a configuration $T = \times_{i \in N} T_i$ (for $T_i \subset S_i$) is wCURB if and only if there is no arc in $\mathcal{G}(G)$ that leaves T.[3]

Next, we show that the intersection of two wCURB sets forms a wCURB set. A similiar result has been shown by Benisch et al. for CURB sets.

Proposition 4 *Let $U = \times_{i \in N} U_i$ and $V = \times_{i \in N} V_i$ with $U_i, V_i \subset S_i$ be two wCURB configurations of the linear extension $\tilde{G}$ of a finite game. Then $T = \times_{i \in N} T_i$ with $T_i = U_i \cap V_i$ is also a wCURB configuration.*

Proof. We fix an arbitrary player $i \in N$ and consider $\beta_i(t)$ for a $t \in T_{-i} = U_{-i} \cap V_{-i}$. As $t \in U_{-i}$ and $t \in V_{-i}$ and by the wCURB property of U and V the best reply $\beta_i(t)$ is in U_{-i} and V_{-i} and hence in $T = U \cap V$. ■

As a consequence of Proposition 4 we obtain that minimal wCURB configurations do not intersect.

4.1 Step one: Finding all wCURB configurations

In this section we present the sub-algorithm that implements step one, that is, identifies all minimal wCURB configurations for any finite game. To this end, let $P(s)$ denote the set of strategy profiles that can be reached by a directed path in $\mathcal{G}(G)$ starting in s. We identify these sets by way of computing, for each pair of strategy profiles $s, t \in S$ the "directed

[3]In graph theory, this condition writes as $\delta_+(T) = \varnothing$.

distance" measured as the number of arcs in a shortest directed path from s to t. We assign the distance $= \infty$ if no such path exists. The computation of these distances can be made in $\mathcal{O}(m^3)$ with the algorithm of Floyd and Warshall, see Cormen et al. (2001) for a description.

Clearly, $P(s)$ contains all vertices t within finite directed distance from s. While $P(s)$ need not to be of the product form $\times_{i\in N}T_i$ (for $T_i \subset S_i$), the algorithm identifies the minimal wCURB configuration, $T(s) = \times_{i\in N}T_i(s)$, that contains $P(s)$. This procedure is initialized by setting $B^0(s) = P(s)$ and letting $T^0(s)$ be the minimal product set, $\times_{i\in N}T_i$, that contains $B^0(s)$. Then let $B^1(s) = \bigcup_{t\in T^0(s)} P(t)$. If $B^1(s) = B^0(s)$, then we have found a minimal wCURB configuration that contains $P(s)$, namely $T^0(s)$. If $B^1(s) \neq B^0(s)$, then let $T^1(s)$ be the minimal product set, $\times_{i\in N}T_i$, that contains $B^1(s)$, and set $B^2(s) = \bigcup_{t\in T^1(s)} P(t)$. If $B^2(s) = B^1(s)$, then $T^1(s)$ is a minimal wCURB configuration that contains $P(s)$. The algorithm proceeds in the same way until $B^{k+1}(s) = B^k(s)$ for a positive integer k. The game being finite, this procedure halts in a finite number of rounds. The pseudocode of this algorithm can be found in Algorithm 1.

Input: finite game G, sets of directedly reachable profiles $(P(s))_{s\in S}$
Output: the family $\mathcal{C}$ of all minimal wCURB configurations in G

foreach $s \in S$ **do**
    $T(s) := \times_{i\in N} \bigcup_{t\in P(s)} \mathrm{supp}(t)$;
end
foreach $s \in S$ **do**
    $C_s := T(s)$;
    `converged` := `false` ;
    while $\neg$ *converged* **do**
        `converged` := `true` ;
        foreach $t \in C_s$ **do**
            if $\neg(T(t) \subset C_s)$ **then**
                $C_s := C_s \cup T(t)$;
                `converged` := `false` ;
            end
        end
    end
    if $\nexists C' \in \mathcal{C} : C' \subset C_s$ **then**
        $\mathcal{C} := \mathcal{C} \cup \{C_s\}$;
        if $\exists C' \in \mathcal{C} : C_s \subsetneq C'$ **then**
            $\mathcal{C} := \mathcal{C} \setminus \{C'\}$;
        end
    end
end

Algorithm 1: Computation of all minimal pure CURB configurations

Proposition 5 *Let a finite game $G = (N, S, u)$ and the sets $(P(s))_{s\in S}$ be given. Algorithm 1 computes all minimal wCURB configurations of G in $\mathcal{O}(m^3)$.*

Proof. The main loop is called once for each $s \in S$ and thus m times. We remark that $T(t)$ can be implemented by a vector of boolean values for each player. During the iteration of the while loop every $T(t), t \in S$ is at most once tested and added to C_s. This can be done in $\mathcal{O}(\max_{i\in N} m_i\, n\, m) \subset \mathcal{O}(m^2)$, delivering the result. ■

As a corollary of this result we obtain that all wCURB sets, of any finite game, can be computed in $\mathcal{O}(m^3)$.

4.2 The full algorithm

Having described the first step of the algorithm, we proceed to present the full algorithm, as applied to the linear extension $\tilde{G}$ of any finite game G. It takes as input a family $\mathcal{T}$ of sets $T = \times_{i\in N} T_i$ (for $T_i \subset S_i$) that is sub-complete with respect to sCURB sets, and gives as output all minimal sCURB sets of the game. The algorithm is general in the sense that the family $\mathcal{T}$ is arbitrary, provided that it satisfies sub-completeness. However, runtime comparisons show that the choice of $\mathcal{T}$ determines the performance of the algorithm considerably. If we would take as input the family $\mathcal{T}$ consisting of all sets $T = \times_{i\in N} T_i$ (for $T_i \subset S_i$), then the algorithm would be equivalent to that of Benisch et al. So the main advantage of the present algorithm is to instead initiate it from the much smaller family of wCURB configurations, as described above. The pseudocode is shown in Algorithm.

In order to prove the correctness of this algorithm, we need the following lemma. We denote by $\mathcal{C}$ the set of wCURB sets that have already been determined.

Lemma 6 *During the run of the algorithm the candidate set $\mathcal{T}$ is sub-complete with respect to remaining sCURB sets, i.e. for every minimal CURB set $C \notin \mathcal{C}$ there is a candidate set $T \in \mathcal{T}$ such that $T \subset C$.*

Proof. By induction. Let k be the number of iterations of the while loop.
"$k = 0$": The completeness property is required when the algorithm starts.
"$k \to k + 1$": Let the completeness property be valid for the first k iterations. During the $k + 1$:st iteration one can distinguish between the two cases: First, suppose in the $k + 1$:st iteration a new sCURB set is found. Then, there can be no minimal sCURB set left that

Input: linear extension $\tilde{G}$ of a finite game, a class of sets $\mathcal{T}$ that is sub-complete with respect to sCURB sets

Output: the set of all CURB sets $\mathcal{C}$

```
1  while T ≠ ∅ do
2  | Choose a size-minimal T ∈ T;
3  | foreach i ∈ N do
4  | | foreach s_i ∉ T_i do
5  | | | if B̂_{is_i} ∩ □(T) ≠ ∅ then
6  | | | | Update all T ∈ T ;
7  | | | | Goto line 2 ;
8  | | | end
9  | | end
10 | end
11 | T := T \ {T} ;
12 | C := C ∪ {C} ;
13 | Update all T ∈ T ;
14 end
```

Algorithm 2: An generic algorithm for sCURB sets in linear extensions of finite games

intersects with the sCURB set found and thus all candidate sets that intersect with the sCURB set found may be erased from the set of candidates $\mathcal{T}$. Second, if a set $T \in \mathcal{T}$ that is minimal in size is chosen and it exists $i \in N$ and $s_i \in S_i$ such that $B_{is_i} \cap \Box(T) \neq \varnothing$. Then there can be no sCURB set that contains the strategies in T_{-i} but not strategy s_i. Thus all candidate sets that contain the strategies in T_{-i} can be extended by s_i and hence the candidate set stays complete. ■

Proposition 7 *Let $\tilde{G}$ be the linear extension of a finite game. Algorithm 2 on page 14 terminates after finitely many steps and identifies all minimal sCURB sets of $\tilde{G}$.*

Proof. In each iteration either a new constraint for the candidate sets or a new sCURB set is found. Every new sCURB set prohibits the played strategies to be included in further candidate sets. If a new constraint is found at least one candidate set, i.e. the candidate set that was checked to be sCURB is enlarged to s_i. So in every iteration at least one set is removed from the set of candidates or at least one set is enlarged about at least one strategy. Thus after finite time either the set of candidates is empty or each candidate is arbitrary large, i.e. the candidate sets are equal to S. As every strategy is played in S the set is recognized to be sCURB. Then every other set that is equal to S is removed from the set of candidates in line 14. Thus the algorithm terminates after finite time. Since the set of candidates $\mathcal{T}$ is complete in every iteration by Lemma 6 and $\mathcal{T} = \varnothing$ after termination, all minimal sCURB sets have been computed. ■

5 Runtime

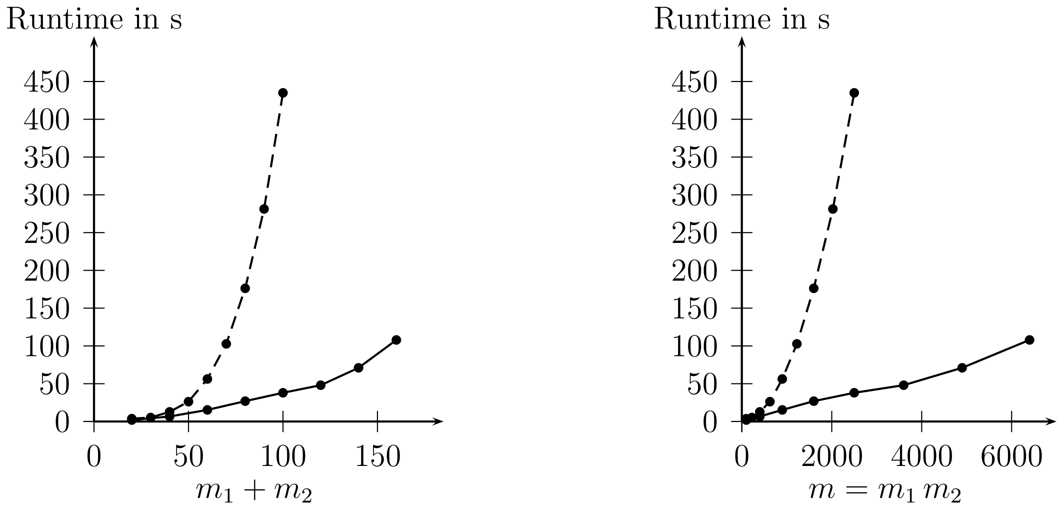
In the implementation of our algorithm, we use SCIP as LP Solver. SCIP is a framework for solving integer and mixed programs developed by Achterberg (2007) in his Ph.D. thesis. It is considered to be one of the fastest non-commercial frameworks to solve mixed integer problems. We use SoPlex in version 1.32 as the underlying simplex algorithm. SoPlex has been developed as part of the Ph.D. thesis of Wunderlich (1996). The whole package can be found at `http://scip.zib.de/`.

For the sake of benchmark comparisons, we computed all minimal sCURB sets in games with random payoffs, generated by GAMUT. The GAMUT framework has become standard for testing game theoretic algorithms (for a detailed description, see Nudelman et al. (2004)). As the two concepts of minimal CURB sets and sCURB sets coincide in two-player games our runtimes can be meaningfully compared with those of the Benisch-Davis algorithm. The results of the time measurements are shown in Figure 4. The dashed line for the Benisch-Davis-Sandholm algorithm is taken from Benisch et al. (2006). The solid line shows the median over 200 games of the time needed to compute all minimal CURB sets. In Figure 4 (a), we plot the average runtime as a function of the total number of pure strategies, $m_1 + m_2$ (as in Benisch et al. (2006)). In Figure 4 (b), we plot the average runtime as a function of the number of pure-strategy profiles, $m = m_1 m_2$. The latter graph shows more directly the "effort" of the algorithm, since this depends on the number of strategy profiles. This plot suggests that the average runtime for our algorithm is nearly linearly for games with up to 6000 strategy profiles, while the runtime of the Benisch et al. algorithm seems to increase exponentially in the number of strategy profiles.

To study the performance of our algorithm when solving games with more than two players, we applied it to games with n players, for $n = 2, 3, ...$, in which each player has only two pure strategies. The average runtime of the algorithm is shown in Figure 5 (a) as a function of the number n of players and in Figure 5 (b) as a function of the number of pure-strategy profiles. These plots show that the time needed to identify all minimal sCURB sets is "affordable" up to $n = 10$ players. In games with more players, the algorithm does not terminate within reasonable time.

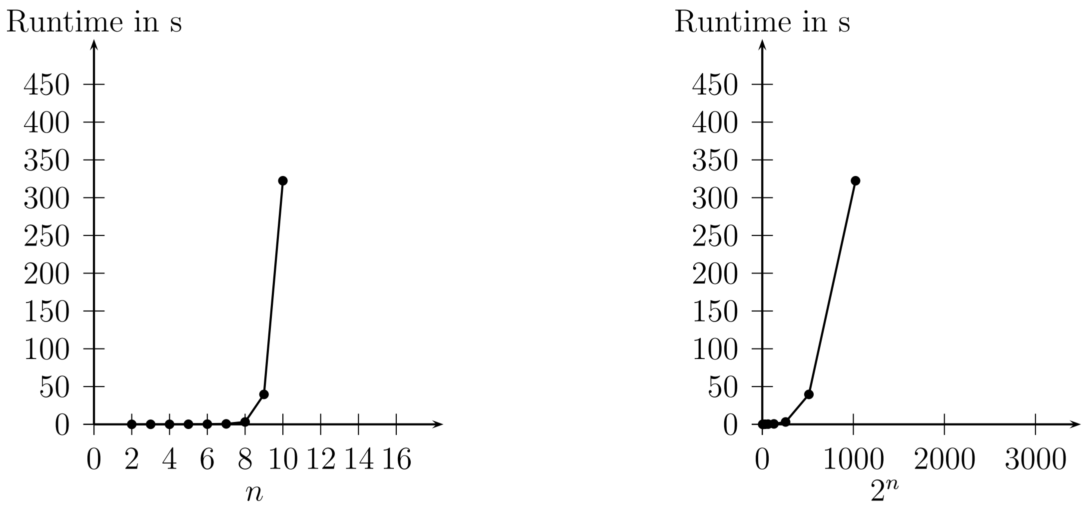
6 Size distribution of minimal CURB sets

Some games have small CURB sets, even singletons, while others have only large CURB sets. For the purpose of prediction, the typical size of CURB sets is of importance. Benisch



a) **b)**

Figure 4: Runtime of algorithms for the computation of all minimal CURB sets in random two-player games a) with respect to $m_1 + m_2$, b) with respect to $m = m_1 m_2$



a) **b)**

Figure 5: Runtime of our algorithm to compute all minimal sCURB sets in random games with n players and two strategies per player a) with respect n, b) with respect to 2^n

et al. (2006) analyze the size distribution of minimal CURB sets in certain classes of random two-player games, generated by GAMUT. They define the size of a minimal CURB set $X = \Delta(T_1) \times \Delta(T_2)$ as the sum of pure strategies involved, $|T_1| + |T_2|$. Hence, for an $m_1 \times m_2$-game, minimal CURB sets range in size from 2 (singletons) to $m_1 + m_2$ (the full strategy space). For each of the randomly generated games in their study, Benisch et al. recorded the smallest size, so defined, of all its minimal CURB sets. They found that the smallest minimal CURB set in a random game, in the studied game lasses, is rarely of intermediate size. As shown in Figure 6, also we find that most games have a smallest minimal CURB set that is either a singleton, corresponding to a strict Nash equilibrium, or it takes up the whole strategy space. Moreover, the proportion of games in which the smallest minimal CURB set is a singleton seems to converge as the number of pure strategies increases. This observation is in agreement with a theoretical result due to Dresher (1970), namely, that the probability that a random $m_1 \times m_2$ game will have at least one pure-strategy Nash equilibrium converges to $1 - 1/e$ as $\min\{m_1, m_2\} \to +\infty$. Formally:

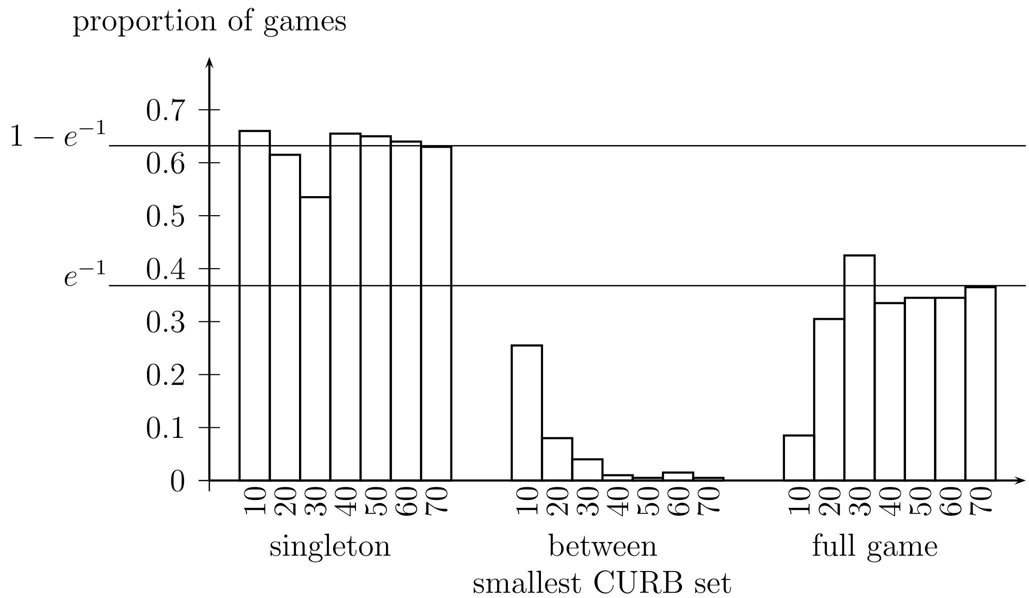


Figure 6: Distribution of smallest CURB set size among 200 randomly drawn games for different game sizes from 10×10 up to 70×70

Theorem 8 (Dresher (1979)) *Let $G = (N, S, u)$ be a finite game where payoff is drawn with the following rules:*

1. *The $n \cdot m$ payoffs are independent random variables*

2. *For each player $i \in N$ the m payoffs $u_i(s)$ have the same continuous probability distribution.*

Then the probability for the existence of at least one pure-strategy Nash equilibrium converges to $1 - e^{-1}$ *as* $\min\{m_i, m_j\} \to \infty$ *for at least two players* i *and* j*.*

In the studied class of random games, a pure-strategy Nash equilibrium is strict with probability one, since payoff ties have zero probability. Hence, each pure-strategy Nash equilibrium, viewed as a singleton set, is, with probability one, a minimal CURB set. Thus, Dresher's theorem implies that we should expect the share of games with smallest minimal CURB set size 2 should converge to $1 - e^{-1}$ as the number of pure strategies tends to infinity:

Corollary 9 *In the linear extension of a finite random game as above the probability that the smallest minimal CURB set is a singleton converges to* $1 - e^{-1}$ *as* $\min\{m_i, m_j\} \to \infty$ *for at least two players* i *and* j*.*

This theoretical result seems to fit well the empirical size distribution in Figure 6; see the thin horizontal line that indicates $1 - e^{-1} \approx 0.632$.

So far, we have studied only the size of the smallest CURB sets. What about the complete size distribution for minimal CURB sets? In Figure 7 we report some results. In panel (a), it is shown that about 90% of the minimal CURB sets in random 2×2-games are singletons. In panel (b), we see that, in random 4×4-games, about 77% of the minimal CURB sets are singletons, while, for example, about 12% of the minimal CURB sets are of size $4 = 2 + 2$, about 4% of size $6 = 3 + 3$ and about 2.5% of size $5 = 2 + 3 = 3 + 2$, etc. In panel (c), we see that minimal CURB sets of intermediate size are not frequent in random games of size 16×16, and in panel (d) we see that they are quite rare in random games of size 32×32. This tendency, towards singletons or the full strategy space, seem to hold for even larger games.

References

T. Abbott, D. Kane, and P. Valiant. On the complexity of two-player win-lose games. *46th Annual IEEE Symposium on Foundations of Computer Science*, pages 113–122, 2005.

T. Achterberg. *Constraint Integer Programming.* PhD thesis, Technische Universität Berlin, July 2007.

G. Asheim, M. Voorneveld, and J. W. Weibull. Epistemic robustness of sets closed under rational behavior. Technical report, Stockholm School of Economics, 2009.

	1	2
1	89.84	
2		10.16

a)

	1	2	3	4
1	77.44			
2		12.40	1.24	
3		1.24	4.34	1.09
4		0.08	1.01	1.16

b)

	1	2	3	4	...	14	15	16
1	74.19							
2		3.32	0.35	0.07				
3		0.21		0.14				
4				0.07				
⋮								
14						0.07		0.21
15							0.35	1.98
16						0.07	1.27	17.68

c)

	1	2	3	...	32
1	71.58				
2		1.85	2.22		
3		0.14	0.07		
⋮					
32					26.13

d)

Figure 7: Distribution of the sizes of minimal sCURB sets a) in random 2×2 games, b) in random 4×4 games, c) in random 16×16 games, d) in random 32×32 games

R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, 1:67 –96, 1974.

R. J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica*, 55(1):1–18, 1987. ISSN 00129682.

K. Basu and J. W. Weibull. Strategy subsets closed under rational behavior. *Economics Letters*, 36:141–146, 1991.

M. Benisch, G. Davis, and T. Sandholm. Algorithms for rationalizability and CURB sets. *National Conf. on Artificial Intelligence*, 2006.

X. Chen and X. Deng. Settling the complexity of two-player nash equilibrium. *47th Annual IEEE Symposium on Foundations of Computer Science*, pages 261–272, 2006.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 2001.

M. Dresher. Probability of a pure equilibrium point in n-person games. *Journal of Combinatorial Theory*, (8):134 –145, 1970.

J. C. Harsanyi and R. Selten. *A General Theory of Equilibrium Selection in Games*. The MIT Press, June 1988.

S. Hurkens. Learning by forgetful players. *Games and Economic Behavior*, (11):304 –329, 1995.

E. Kalai and D. Samet. Persistent equilibria in strategic games. *International Journal of Game Theory*, 13(3):129 – 144, 1984.

M. Klimm. Algorithms for curb sets. Technical report, École Polytechnique, 2008.

E. Kohlberg and J. F. Mertens. *Econometrica*, 54:1003 – 1037, 1986.

C. E. Lemke and J. J. T. Howson. Equilibrium points of bimatrix games. 12(2):413 – 423.

R. B. Myerson. Refinements of the Nash equilibrium concept. *International Journal of Game Theory*, 7(2):73 – 80, 1978.

E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 880–887, 2004.

D. Pearce. Rationalizable strategic behavior and the problem of perfection. *Econometrica*, 52:1029 – 1050, 1984.

V. Pruzhansky. On finding curb sets in extensive games. *International Journal of Game Theory*, 32(2):205 – 210, 2003.

R. Savani and B. von Stengel. Exponentially many steps for finding a nash equilibrium in a bimatrix game. *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 258–267, Oct. 2004.

R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *International Journal of Game Theory*, 4(1):25 – 55, 1975.

M. Voorneveld. Preparation. *Games and Economic Behavior*, (48):403 – 414, 2004.

M. Voorneveld. Persistent retracts and preparation. *Games and Economic Behavior*, (51): 228 – 232, 2005.

R. Wunderlich. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996.

Y. Ye. Improved complexity results on solving real-number linear feasibility problems. *Mathematical Programming*, 106:339 – 363, 2006.

H. P. Young. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton University Press, Princeton, NJ, USA, 1998.